CLAIMS

What is claimed is:

1. An instruction bundle comprising:

a first instruction; and

a second instruction, wherein the second instruction includes a pointer to an exception handler and wherein the second instruction includes a predetermined offset from the first instruction.

2. The instruction bundle of claim 1, wherein the predetermined offset includes a predetermined number of instructions between the first instruction and the second instruction. 3.The instruction bundle of claim 1, wherein the second instruction includes a plurality of non-executable bits.

4. The instruction bundle of claim 1, wherein the second instruction is a no op instruction.

5. The instruction bundle of claim 1, wherein the second instruction precedes the first instruction.

6. The instruction bundle of claim 1, wherein the first instruction precedes the second instruction.

7. The instruction bundle of claim 1, wherein the exception handler includes a plurality of exception-type handlers.

8. The instruction bundle of claim 1, wherein the instruction bundle is an IA64 instruction bundle.

9. The instruction bundle of claim 1, wherein the instruction bundle is included in an instruction set that includes at least one instruction in the instruction bundle with an unused bit space.

10. An instruction bundle comprising:

a first instruction; and

a no op instruction, wherein the no op instruction includes a pointer to an exception handler and wherein the no op instruction is offset a predetermined number of instructions from the first instruction.

11. The instruction bundle of claim 10, wherein the instruction bundle is an IA64 instruction bundle.

12. A method for handling an exception comprising:

executing a first instruction wherein the first instruction returns an exception; and

using a program counter to determine the location of a second instruction, wherein the second instruction includes a pointer to an exception handler.

13. The method of claim 12, wherein the pointer includes a number of instructions between a subsequent instruction and the exception handler, wherein the subsequent instruction follows the first instruction.

14. The method of claim 12, wherein the pointer includes an encoded address of the exception handler.

15. The method of claim 12, wherein the pointer includes a number of instructions between the first instruction and the exception handler.

16. The method of claim 12, wherein the pointer includes a number of instructions between the second instruction and the exception handler.

17. The method of claim 12, wherein the second instruction is offset a predetermined number of instructions from the first instruction.

18. The method of claim 12, wherein the program counter points to a subsequent instruction that a program execution stack will return to after the first instruction is executed.

19. The method of claim 12, wherein the exception handler includes a plurality of types of exception handlers.

20. The method of claim 19, further comprising:

determining a type of the exception; and

determining if one of the plurality of types of exception handlers matches the type of the exception.

21. The method of claim 12, wherein the first instruction and the second instruction are included in an instruction bundle.

22. The method of claim 12, wherein using a program counter to determine the location of the second instruction includes identifying the second instruction as an instruction that is a predetermined number of instructions from a return instruction that is pointed to by the program counter.

23. A compiled code segment comprising:

a first instruction; and

a no op instruction, wherein the no op instruction is a fixed number of instructions from the first instruction and wherein the no op instruction includes a pointer to an exception handler.

24. A compiled code segment comprising:

a first return sequence that returns normally; and

a return sequence that includes a program counter, wherein the program counter indirectly points to an exception handler.

25. A compiled code segment of claim 24 wherein the program counter points to a second instruction and wherein the second instruction includes a pointer to the exception handler.

26. A system for processing an exception comprising:

a processor;

a storage facility coupled to the processor and containing instructions executable by the processor which configure the system to:

execute a first instruction wherein the first instruction returns an exception; and

use a program counter to determine the location of a second instruction, wherein the second instruction includes a pointer to an exception handler.

27. A system of claim 26, wherein the pointer includes an encoded address of the exception handler.